REMARKS

Claims 1, 2, 5-24, 29-59, 61 and 62 are presently pending in the instant application. In the final Office Action dated 5/8/02, the Examiner rejected all pending claims. In a telephone interview of June 20, 2002, with Applicants' counsel, the Examiner indicated that he would reconsider his rejection of the claims, and in particular independent claims 29, 30 and 61. With this Amendment, claims 59 and 61 are amended. Applicants respectfully request reconsideration, reexamination and allowance of the application.

On page 2 of the Office Action, the Examiner rejected claim 59 under 35 U.S.C. 112 for depending on cancelled claim 25. To alleviate this situation, claim 59 is amended herewith to be in independent form. Also, claim 61 is amended herewith to correct a spelling error.

The Examiner also rejected claims 1, 2, 5, 19 and 62 under 35 U.S.C. §102 as allegedly being unpatentable over Ben-Yoseph et al., U.S. Patent 5,949,439. Applicants respectfully submit that claim 1 amended is not anticipated by the Ben-Yoseph patent because, among other reasons, Ben-Yoseph does not disclose at least the claimed combination wherein the coprocessor processes the plurality of components of the pixel in parallel as elements of a vector. On pages 2 and 3 of the Office Action, the Examiner asserts that Ben-Yoseph teaches that the coprocessor processes the plurality of components of the pixel in parallel as elements of a vector. Applicants respectfully disagree. Ben-Yoseph discloses a multimedia processor 106 that uses Very Long Instruction Word (VLIW), vector processing, and single-instruction multiple data (SIMD) technology to achieve parallel operation. Col. 3, lns. 35-39. The multimedia processor 106 transmits and receives data simultaneously over a high-speed Rambus DRAM channel I/O bus. Col. 3, lns. 43-45. But to say that the multimedia processor of Ben-Yoseph achieves parallel operation and transmits and receives data simultaneously over a DRAM channel I/O bus is not to say that Ben-Yoseph teaches that the coprocessor processes the plurality of components of a pixel in parallel as elements of a vector, as is claimed in amended claim 1. Ben-Yoseph, while broadly disclosing a multimedia processor that achieves parallel processing, does not provide much if any detail as to what types of data elements are processed in parallel. The term "parallel processing,"

E

when used in the broad sense that Ben-Yoseph uses it, could refer to any of a plethora of things. Such a broad disclosure of a multimedia processor that achieves parallel processing does not foreclose the patenting of a new and inventive use of parallel processing in a graphics accelerator. Claim 1's processing of the components (such as the R, G and B components) of a pixel in parallel is such a new and inventive use of parallel processing. There is simply no disclosure in Ben-Yoseph of the processor 106 processing a plurality of components of a pixel in parallel as elements of a vector. As neither Ben-Yoseph nor Gulick discloses at least this element of amended claim 1, Applicants respectfully request allowance of claim 1 and claims dependent thereon.

On page 3 of the Office Action, the Examiner rejected claims 23, 29-31, 37, 44-46, 50, 51 and 61 under 35 U.S.C. 103(a) as being allegedly unpatentable over Ben-Yoseph. Claim 23 is amended herewith for clarification. Claim 23 as amended includes a direct memory access (DMA) engine for transferring the graphics data between an external memory and the local memory, wherein the DMA engine moves data between the local memory and the external memory while the graphics accelerator is using the local memory for its load and store operations. Applicants submit that while Ben-Yoseph discloses using DMA as a means of transferring data between external memory and a processor (or SRAM associated with a processor), nowhere does Ben-Yoseph disclose a DMA engine that moves data between the local memory and the external memory at the same time the graphics accelerator is using the memory for load and store operations.

On page 4 of the Office Action, the Examiner concedes that Ben-Yoseph does not explicitly teach this aspect of the present invention. But the Examiner then argues that "it would have been obvious to one of ordinary skill in the art… to implement the teachings of Ben-Yoseph as claimed because Ben-Yoseph et al teaches the multimedia processor 106 operates under control of a real time multitasking kernel and simultaneously addresses the five semi-independent, quasi-specialized execution units 152; transmits and receives data simultaneously over a high speed RAMbus bus DRAM channel I/O bus and other buses, no wait to send and received (sic) data between peripherals, the system bus, or the Rambus DRAM 110 (col. 3, lns. 40-56 and col. 5, lns. 34-40) which clearly shows simultaneous operations as claimed." The excerpt cited by the Examiner contains no mention of a DMA engine nor any mention of any components that could be considered

analogous to a DMA engine. It is not clear to Applicants what element of the cited excerpt the Examiner deems to be, or to be equivalent to, a DMA engine. Applicants submit that the multimedia processor 106 of Ben-Yoseph is not a DMA engine nor is it analogous in any way to a DMA engine. Furthermore, even if the multimedia processor 106 were a DMA engine, the multimedia processor does not serve to move data between a local memory and an external memory while a graphics accelerator is using the local memory for its load and store operations. The fact that the multimedia processor 106 can simultaneously transmit and receive data in no way renders claim 23 obvious. Thus, Applicants respectfully request allowance of claim 23 and claims dependent thereon.

Claim 29 includes a direct memory access (DMA) engine for transferring the graphics data between an external memory and the local memory, wherein the graphics accelerator is working on operands and producing outputs for one set of pixels, while the DMA engine is bringing in operands for a future set of pixel operations. On page 4 of the Office Action, the Examiner states that claim 29 would have been obvious for the same reason set forth above with respect to claim 23. With respect to claim 23, the Examiner argued that "it would have been obvious to one of ordinary skill in the art… to implement the teachings of Ben-Yoseph as claimed because Ben-Yoseph et al teaches the multimedia processor 106 operates under control of a real time multitasking kernel and simultaneously addresses the five semi-independent, quasi-specialized execution units 152; transmits and receives data simultaneously over a high speed RAMbus bus DRAM channel I/O bus and other buses, no wait to send and received (sic) data between peripherals, the system bus, or the Rambus DRAM 110 (col. 3, lns 40-56 and col. 5, lns. 34-40) which clearly shows simultaneous operations as claimed." The excerpts cited by the Examiner with respect to claim 23 contain no mention of a DMA engine nor any mention of any components that could be considered analogous to a DMA engine. It is not clear to Applicants what element of the cited excerpts the Examiner deems to be, or to be equivalent to, a DMA engine. Ben-Yoseph does refer to DMA, at col. 4, line 62 – col. 5, line 4, but the excerpts cited by the examiner do not refer to DMA. Applicants submit that the multimedia processor 106 of Ben-Yoseph is not a DMA engine nor is it analogous in any way to a DMA engine. Furthermore, even if the multimedia processor 106 were a DMA engine, there is no indication in Ben-Yoseph that the multimedia

E

processor works on operands and produces outputs for one set of pixels, while bringing in operands for a future set of pixel operations. The fact that the multimedia processor 106 can simultaneously transmit and receive data in no way renders claim 29 obvious. Additionally, the portions of Ben-Yoseph that do mention DMA refer in no way to the multimedia processor 106 working on operands and producing outputs for one set of pixels, while the DMA brings in operands for a future set of pixel operations.

The Examiner also supports his rejection of claim 29 on page 10 of the Office Action by stating that Ben-Yoseph's SRAM 146 is multiported. However, as the Examiner pointed out, "multiported" simply means that the SRAM can be accessed by multiple modules concurrently. "Multiported" in no way implies that the multimedia processor 106 works on operands and produces outputs for one set of pixels, while bringing in operands for a future set of pixel operations. Additionally, Ben-Yoseph's inclusion of a multiported SRAM indicates nothing about a DMA module or its operation. Thus, Applicants respectfully request allowance of claim 29 and claims dependent thereon.

Claim 30 includes the steps of loading a block of graphics data from main memory into local memory of a graphics accelerator having a processor and a coprocessor, the graphics data including pixels, each pixel having a plurality of components; performing operations on the plurality of components of each pixel of graphics data using the coprocessor; concurrently transferring blocks of unprocessed data and processed data between the main memory and the local memory while the block of graphics data is being processed. There is no disclosure in Ben-Yoseph of the claimed method, including the concurrent transferring of graphics data between local memory and main memory as a block of data is being processed. On pages 4 and 5 of the Office Action, the Examiner asserts that this method would have been obvious by the teachings of Ben-Yoseph for the same reason set forth above with respect to claim 23. Applicants respectfully disagree with this assertion. Claim 30's concurrent transferring of graphics data between local memory and main memory as a block of data is being processed is made possible by the DMA engine 1304 that has a DMA command queue 1306. Applicants submit that Ben-Yoseph contains no functionality that in itself would perform the functions of claim 30. The fact that the multimedia processor 106 can simultaneously

E

transmit and receive data does not render obvious claim 30's concurrent transferring of graphics data between local memory and main memory as a block of data is being processed. The Examiner also supports his rejection of claim 29 on page 10 of the Office Action by stating that Ben-Yoseph's SRAM 146 is multiported. However, as the Examiner pointed out, "multiported" simply means that the SRAM can be accessed by multiple modules concurrently. "Multiported" in no way implies that graphics data is transferred between local memory and main memory concurrent to the processing of a block of data. Accordingly, Applicants respectfully request allowance of claim 30 and all claims dependent thereon.

Claim 61 includes steps of processing a first block of graphics data; transferring a second block of unprocessed graphics data from the main memory to the on-board memory while the first block of graphics data is being processed; transferring a third block of processed graphics data from the on-board memory to the main memory while the first block of graphics data is being processed. There is no disclosure in Ben-Yoseph of the claimed method, including the concurrent transferring of graphics data between local memory and main memory as a block of data is being processed. On page 5 of the Office Action, the Examiner asserts that this method would have been obvious by the teachings of Ben-Yoseph for the same reason set forth above with respect to claim 23. Applicants respectfully disagree with this assertion. While Ben-Yoseph teaches simultaneously transmitting and receiving data over the high speed Rambus DRAM channel, it does not disclose the simultaneous processing of graphics data. Claim 61's processing of graphics data while concurrently transferring processed and unprocessed blocks of graphics data between local memory and main memory is made possible by the DMA engine 1304 that has a DMA command queue 1306. Applicants submit that Ben-Yoseph contains no functionality that in itself would perform the functions of claim 61. The fact that the multimedia processor 106 can simultaneously transmit and receive data in no way renders obvious claim 61's processing of graphics data while concurrently transferring processed and unprocessed blocks of graphics data between local memory and main memory. The Examiner also supports his rejection of claim 29 on page 10 of the Office Action by stating that Ben-Yoseph's SRAM 146 is multiported. However, as the Examiner pointed out, "multiported" simply means that the SRAM can be accessed

E

by multiple modules concurrently. "Multiported" in no way implies that graphics data is processed while concurrently transferring processed and unprocessed blocks of graphics data between local memory and main memory. Accordingly, Applicants respectfully request allowance of claim 61 and all claims dependent thereon.

Based on the foregoing, Applicants submit that claims 1, 2, 5-24, 29-59, 61 and 62 are in condition for allowance. Applicants, therefore, respectfully request early issuance of a Notice of Allowance.

Date: June 24, 2002

Respectfully submitted,

John A. Wiberg
Reg. No. 44,401

McAndrews, Held & Malloy, Ltd.
500 W. Madison, Suite 3400
Chicago, IL 60661
Ph: 312 775 8000
Fx: 312 775 8100

E

VERSION WITH MARKINGS TO SHOW CHANGES MADE

Claims 59 and 61 are amended as follows:

59. (Twice amended) [The] A graphics accelerator [of claim 25] comprising:

a local memory for storing graphics data;

a coprocessor for performing operations on the graphics data; and

a direct memory access (DMA) engine for transferring the graphics data between an external memory and the local memory, wherein the coprocessor posts a plurality of commands to the queue and the DMA executes the commands concurrently with the operation of the coprocessor.

61. (Amended) A method for sequentially processing blocks of graphics data, the method comprising the steps of:

transferring a first block of unprocessed graphics data from main memory to on-board memory;

processing the [frist] first block of graphics data;

transferring a second block of unprocessed graphics data from main memory to the on-board memory while the first block of graphics data is being processed; and

transferring a third block of processed graphics data from the on-board memory to the main memory while the first block of graphics data is being processed.